H264 ЖИВ

Иван Емельянов
Руководитель разработки видеохостинга
Дзен



Что вы узнаете?

- Как **измерять визуальное качество** для сотен тысяч видео



Что вы узнаете?

- Как **измерять визуальное качество** для сотен тысяч видео
- Какие есть способы **пережатия видео**



Что вы узнаете?

- Как **измерять визуальное качество** для сотен тысяч видео
- Какие есть способы **пережатия видео**
- И главное: как **сэкономить трафик на 30%** (да-да, это мы умеем)

Видео в Дзене

Дзен

500 тыс

авторов,
загрузивших видео
за год

До 1 Тбит/с

видеотрафика

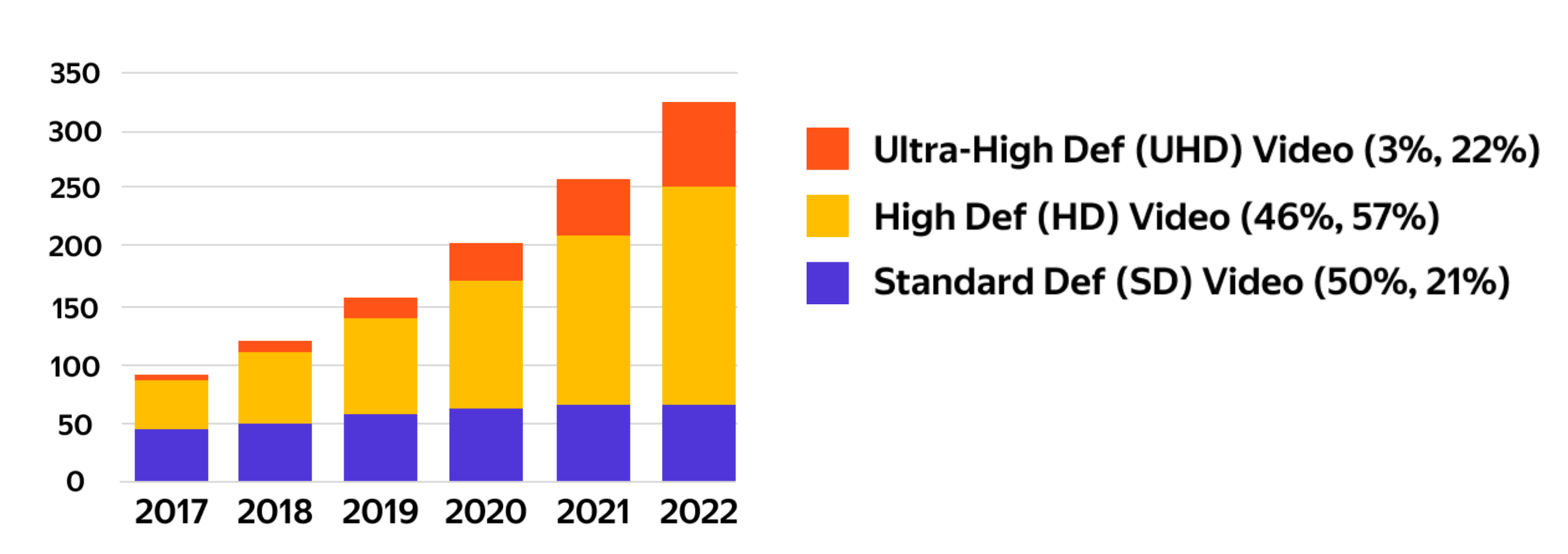
10 млн

видео загружено
за год



Почему это важно?





Меньше физический размер видео



Меньше проблем с буферизацией



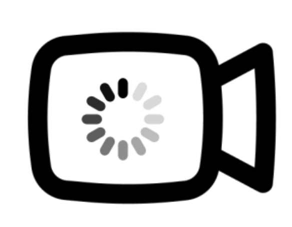
Пользователь доволен и счастлив



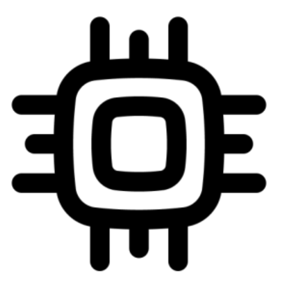
Как работает видеохостинг?



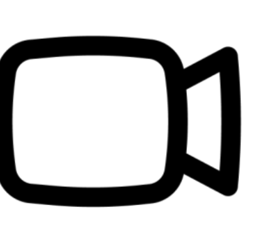
Пользователь →



загрузка видео →



транскодер →



показать видео



Дзен

Как работает транскодирование?

Реализация энкодера

```
ffmpeg -i <input> -c:v libx264 -b:v 1M -maxrate
1M -bufsize 2M -pass 1 -f null /dev/null
ffmpeg -i <input> -c:v libx264 -b:v 1M -maxrate
1M -bufsize 2M -pass 2 <output>
```

Ограничивает битрейт выходного видео

История

MPEG-4
Low-bitrate multimedia and internet

H.264 / AVC
Internet, computers, mobile and HDTVs

1994 **1999** **2003**

MPEG-2
MPEG2 was superior version of MPEG 1

Дзен



История

MPEG-4
Low-bitrate multimedia and internet

H.264 / AVC
Internet, computers, mobile and HDTVs

H.265 / HEVC Ultra
HD to 8K 50% improvement over H.264

AOM is formed and work in the AV1 codec begins














mozilla

1994 **1999** **2003** **2010** **2013** **2015** **2018**

MPEG-2
MPEG2 was superior version of MPEG 1

VP8
Google invests in a new compression standard

VP9
Performs at same level to H.265

VP10
Development is moved to support the AV1 code base

AV1
Code freeze allowing for the first major commercial applications



В чем проблема современных кодеков?

- **H264** поддерживается **везде**

Дзен



В чем проблема современных кодеков?

- **H264** поддерживается **везде**
- **Энкодинг** современных кодеков — **дорого**



В чем проблема современных кодеков?

Дзен

- **H264** поддерживается **везде**
- **Энкодинг** современных кодеков — **дорого**
- Хранить видео в **нескольких кодеках** — тоже дорого



А если не менять кодек?

Дзен

Maxrate, b:v, bufsize — **ограничивают битрейт** выходного видео

```
ffmpeg -i <input> -c:v libx264 -b:v 1M -maxrate
1M -bufsize 2M -pass 1 -f null /dev/null
ffmpeg -i <input> -c:v libx264 -b:v 1M -maxrate
1M -bufsize 2M -pass 2 <output>
```




Метрики визуального качества

Дзен

Безреференсные

- NIQE

Референсные*

- PSNR
- SSIM
- VMAF

*Сравниваем с исходником
и считаем некоторый quality loss





Идея 1

Кодируем в целевое значение метрики



Идея 1: целевые метрики

Дзен

Пример: VMAF 60

Как найти оптимальное значение битрейта?

Бинпоиском перебираем битрейт, пока не найдем оптимальное значение



Идея 1: целевые метрики

Дзен

Разные метрики: SSIM, PSNR, VMAF, VMAF-neg, VMAG with conf interval…



Идея 1: целевые метрики

Дзен

Разные метрики: SSIM, PSNR, VMAF, VMAF-neg, VMAG with conf interval…

Разные агрегации: mean, harmonic mean, avg в окне, percentile

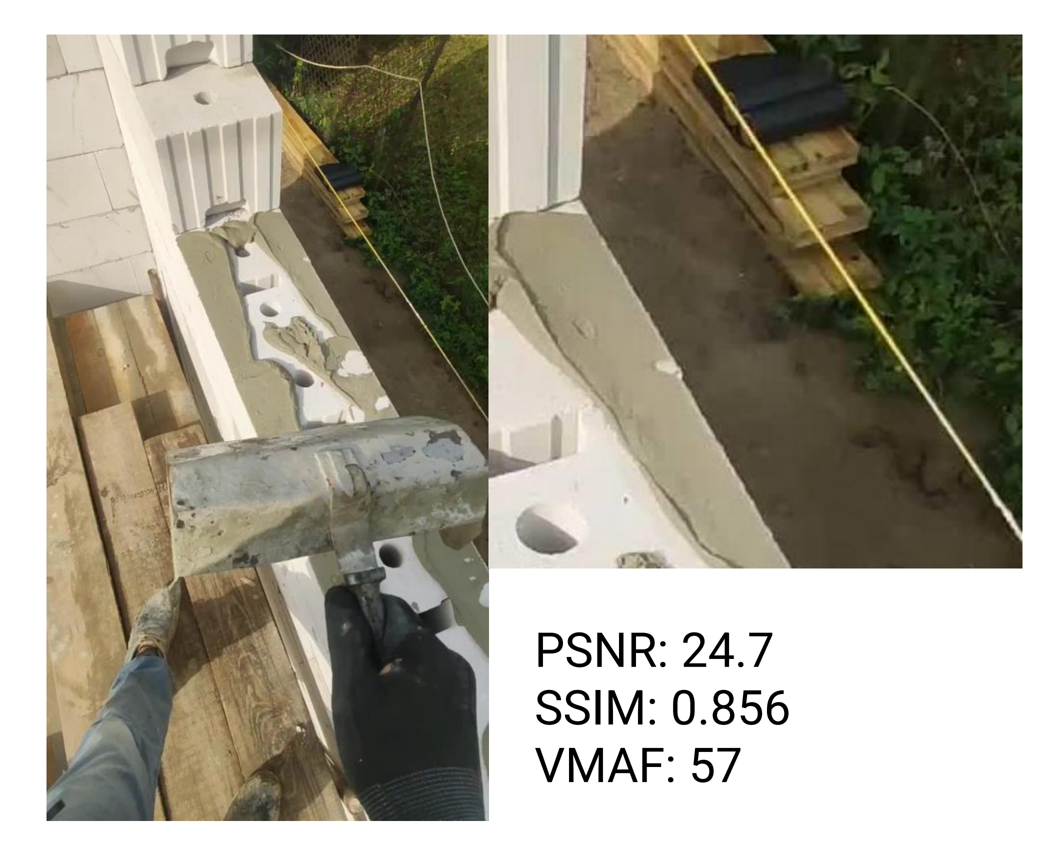


Идея 1: выводы

Метрики плохие, но визуально качество не пострадало



Original, 8977 kbit/sec



Compressed, 7507 kbit/s





Идея 1: выводы

Метрики лучше, а результат заметно хуже:



Original, 11983 kbit/s

PSNR: 36.7
SSIM: 0.941
VMAF: 63

Compressed, 300kbit/s

Идея 1: целевые метрики

Дзен

Вывод: метрики не всегда коррелируют
с восприятием человека



Идея 2

Дзен

Изменить битрейт



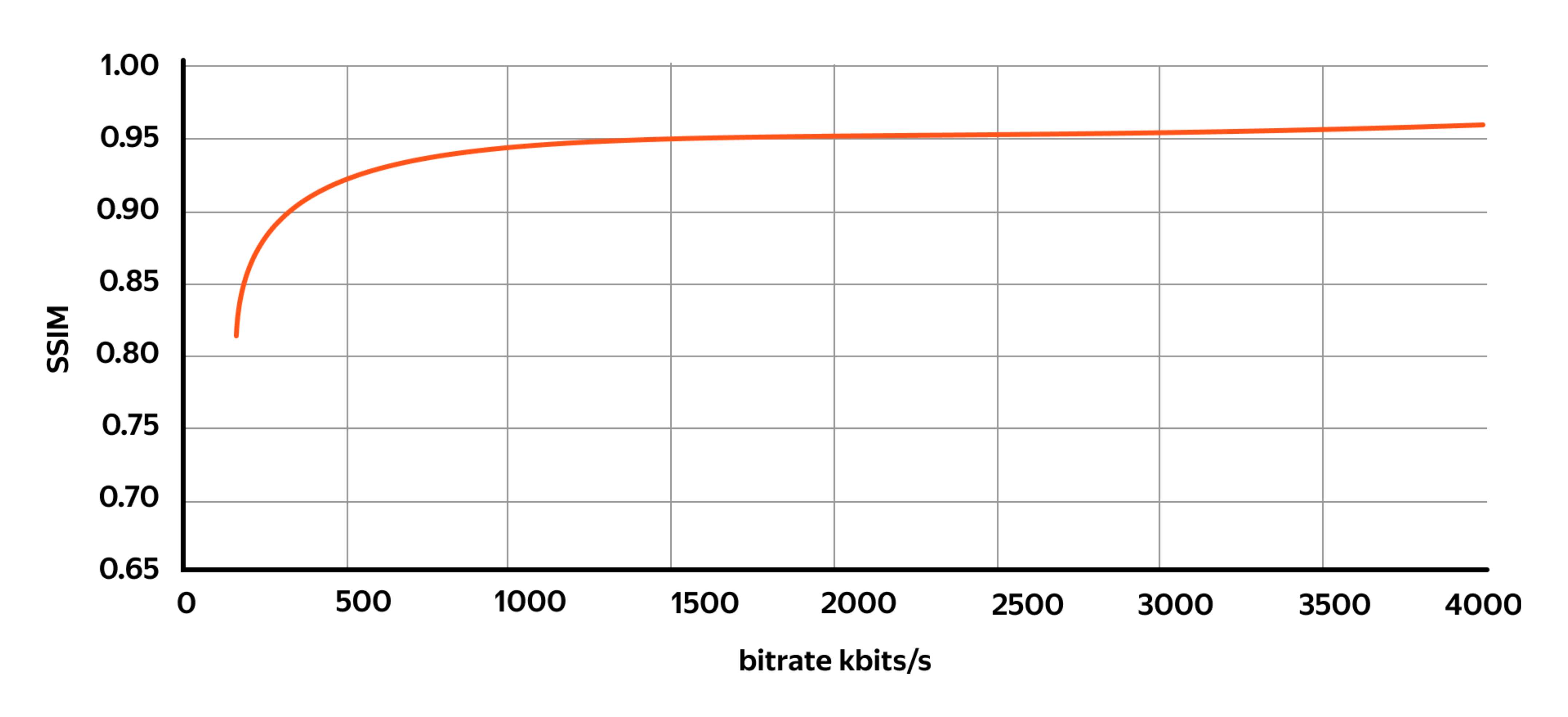


Идея 2: изменить битрейт

А можно ли вообще **уменьшать** только **битрейт** и **не ронять** визуальное **качество**?



Идея 2: изменить битрейт



RD-curve – rate distortion curve. Показываем зависимость визуального качества от битрейта для одного видео

Идея 2: изменить битрейт

Дзен

Вывод: непонятно, сколько «визуального качества» можно потерять, чтобы осталось «хорошо»



Идея 3

А что там в YouTube?





Идея 3: что в YouTube?

Дзен

VMAF our
VMAF youtube

SSIM our
SSIM youtube

PSNR our
PSNR youtube

size our
size youtube

Зависимость Vmaf, Ssim, Psnr, и размера файла от времени



Идея 3: что в YouTube?

- Перебрали параметры
- Те же параметры, но однопроходное кодирование с crf

Пример:

```
ffmpeg -i <input> -c:v libx264 -crf 23 -
maxrate 1M -bufsize 2M <output>
```




Crf vs 2-х проходное кодирование



Crf vs 2-х проходное кодирование

Дзен

- **CRF** — constant rate factor. Все кадры пережимаются «одинаково». Размер итогового файла не гарантирован.



Crf vs 2-х проходное кодирование

- **CRF** — constant rate factor. Все кадры пережимаются на «одинаково». Размер итогового файла не гарантирован.

- **2-pass** — собирается статистика по всему видеофайлу; и кодек, опираясь на эту статистику, пережимает файл.



Crf vs 2-х проходное кодирование

Дзен

- **CRF** — constant rate factor. Все кадры пережимаются на «одинаково». Размер итогового файла не гарантирован.
- **2-pass** — собирается статистика по всему видеофайлу; и кодек, опираясь на эту статистику, пережимает файл.
- **Вывод:** однопроходное кодирование работает в 1,5 раза быстрее, но размер результата не гарантирован.



Как стало

VMAF our
VMAF youtube

SSIM our
SSIM youtube

PSNR our
PSNR youtube

size our
size youtube

Зависимость Vmaf, Ssim, Psnr и размера от времени



Идея 3: что в YouTube?

Как подбирать crf?

Собрать датасет одинаковых видео

Собрать фичи

Обучить модель

Идея 3: что в YouTube?

Как собирать датасет?

- Короткие видео
- Нашли похожие видео на ютубе
- Нашли точные дубликаты через фингерпринты



Идея 3: что в YouTube?

Какие фичи использовать?

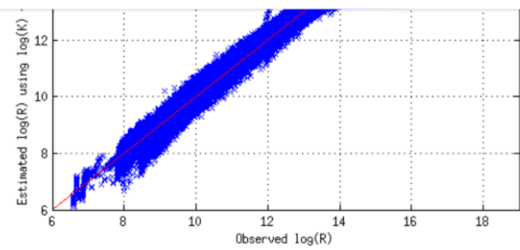


Figure 4: Quality of fit of estimated $\log R(v)$ using NNLS-fit values for $\log K(v)$, $a(v)$, and $d(v)$.
(Pearson's correlation: 0.9984. Error std.: 0.1. Max error: 1.41)

where $R(c,t,h,v)$ is the predicted bitrate for a segment of video v, given the requested CRF setting, c; the requested frame rate, t; the requested frame height, h,[3] and where $K(v)$, $a(v)$, $b(v)$, and $d(v)$ are hidden variables that are dependent only on the video content v.

We tested this relationship on 9,250 5-second video segments, sampled from 1000 of the videos that had been recently uploaded to YouTube. We selected up to 5 different frame heights (from 240 to 1080 but omitting heights that are larger than the original video height) and 29 different CRF settings (from 12 - 40), resulting in up to 145 different samples of each video segment.[4] For each of the 9,250 video segments, we found the best fit for $\log K$, a and d, using the non-negative least squares routine that is included in SciPy [4, 8]. None of the bitrate-model parameters ($\log K$, a, b, and d) can be valid when less than zero.[5] Since Ma et. al [6] had already shown the log-linear relationship between bitrate and frame rate, we did not increase our testing data by repeating that part of the experiment. The blue curves in Figures 1 through 3 show the distributions that we found from the NNLS fit to our training data.

We show the quality of the bitrate estimation in Figure 4

eters as part of a process for estimating the correct CRF to use for our target bitrate using video features that are available to use from earlier processing in the upload pipeline. We discuss this possibility in the next section.

Features from the Mezz

Files uploaded to YouTube are completely unconstrained. Even though unusual filetypes and bitstreams (e.g. variable frame rates, colorspaces, incorrect containers etc) are encountered for a small fraction of these uploads, the volume of ingest means that these edge cases can challenge the robustness of the transcoding system. Therefore, we normalize files before transcoding by creating a high-bitrate, constant-frame-rate mezzanine. We refer to the output of this re-encode as our *mezz*. The process of mezz creation gives us access to encoding properties that characterise the video stream. We collect the following features (accumulated over each video segment) from the creation of the mezz bitstream, for prediction of each segment's bitrate-model parameters:

- average number of bits used for the move vector (MV) per predicted (P) macroblock (MB)
- average number of bits used for the texture (i.e., pixel values or prediction error) per MB
- average number of bits used for the texture per intraframe (I) MB
- average number of bits used for the texture per P MB
- percentage of I MBs
- percentage of skipped MBs
- a score on the complexity of the video encoding for this segment
- average quantization-parameter setting used in the mezz transcode



Есть статья:
Optimizing Transcoder Quality Targets Using a Neural Network with an Embedded Bitrate Model



Идея 3: что в YouTube?

Какие фичи использовать?

Закодировать в базовый вариант с crf и снять статистики:

- **битрейт** на выходе
- % **макроблоков** по типам
- средний размер **макроблоков** каждого типа
- информация о **motion-векторах**
- средний параметр **квантизации макроблоков** каждого типа
- и другие

Идея 3: фичи

А также:

- **Ti — Temporal perceptual information**
  средний размер P-фреймов / средний размер I-фреймов
- **Si** — Spatial perceptual information
  средний размер I-фреймов / максимальный размер I-фреймов



Идея 3: выводы

✦ **CRF** может **ускорить кодирование**

Дзен



Идея 3: выводы

Дзен

- **CRF** может **ускорить кодирование**
- **CRF** можно подбирать для достижения **нужного битрейта**



Идея 3: выводы

✦ **CRF** может **ускорить кодирование**

✦ **CRF** можно подбирать для достижения **нужного битрейта**

✦ Это **может ML**



Идея 3: выводы

Дзен

- **CRF** может **ускорить кодирование**
- **CRF** можно подбирать для достижения **нужного битрейта**
- Это **может ML**
- **Экономия** пока **недостаточно** высокая



Идея 4

Дзен

Как еще можно пережимать видео тем же кодеком, но не терять в визуальном качестве?



Идея 4: пресеты

Почитали статьи — нашли идею:

>70
параметров
x264

>40
параметров
libvp9



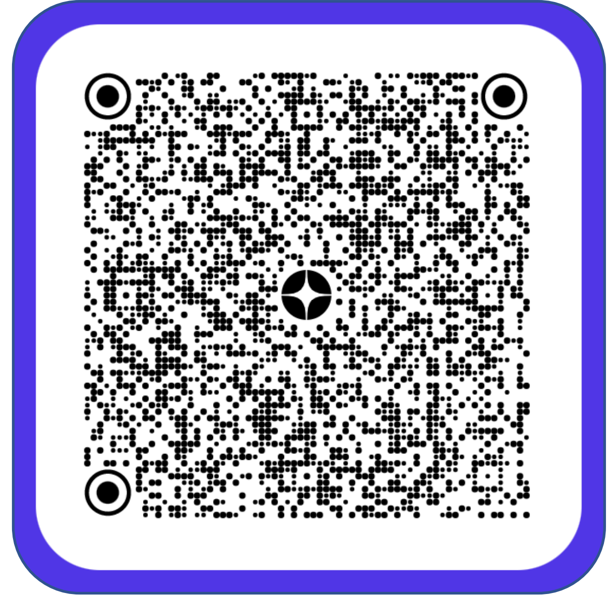
Идея 4: пресеты

Дзен

pareto optimal preset
standard preset
other preset

a relative average encoding time, T(v,p)

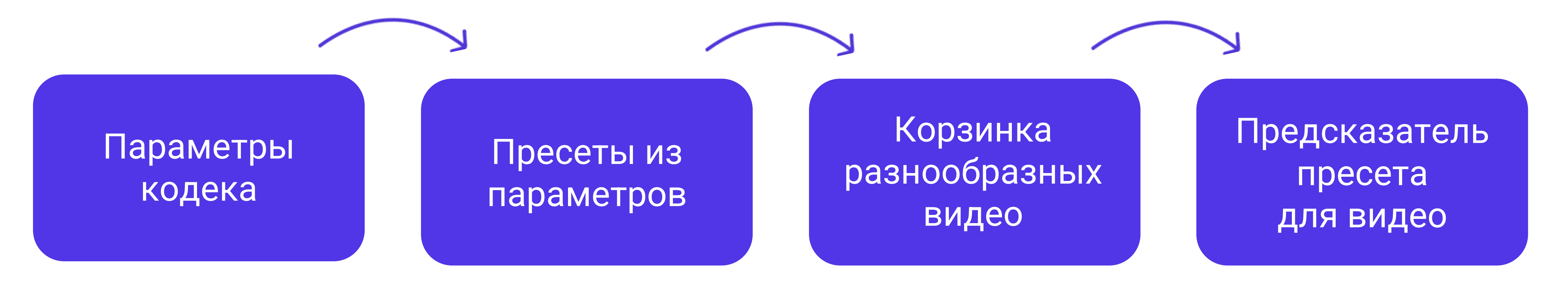
a relative average bitrate, Q(v,p)



Machine-Learning-Based Method for Finding Optimal Video-Codec Configurations Using Physical Input-Video Features

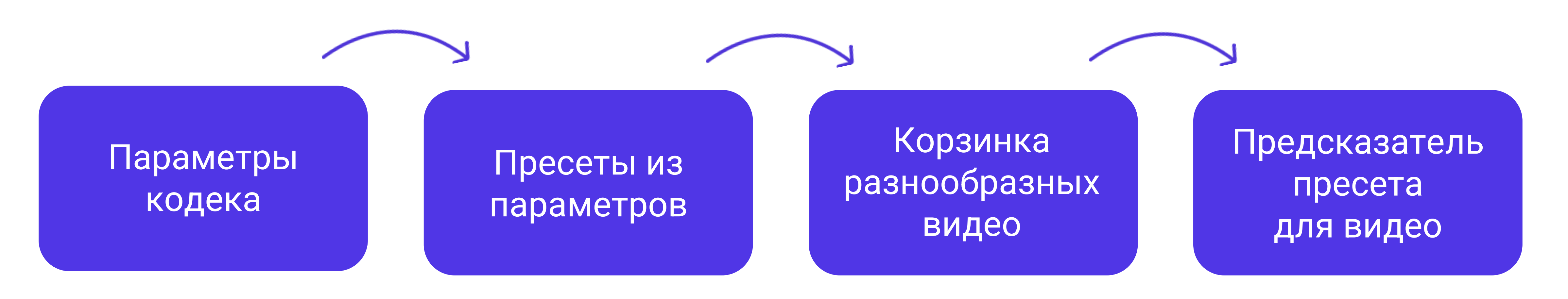
HighLoad++ 2022

Идея 4: пресеты





Идея 4: пресеты



- Собрали **топ-частотные параметры кодека:** статьи, парсинг интернета, стандартные пресеты
- Получили около **50 параметров**



Идея 4: пресеты

- Комбинаций **10^50**
- Попробовали **NSGA-II** для перебора



Идея 4: пресеты

Дзен

- Комбинаций **10^50**
- Попробовали **NSGA-II** для перебора



Итог: умеем эффективно перебирать пресеты для одного видео



NSGA-II

Благодаря **NSGA-II** мы оптимизируем **не один** параметр, **а два**:
- ✦ Выходной битрейт
- ✦ Скорость кодирования



Идея 4: пресеты

Результат кодирования – 3х-мерное пространство

Измерения:



Идея 4: пресеты

Результат кодирования – 3х-мерное пространство

Измерения:

- Итоговый **битрейт**



Идея 4: пресеты

Результат кодирования – 3х-мерное пространство

Измерения:

- Итоговый **битрейт**
- Итоговое **визуальное качество** — примерно одинаковые метрики

Дзен



Идея 4: пресеты

Результат кодирования – 3х-мерное пространство

Измерения:

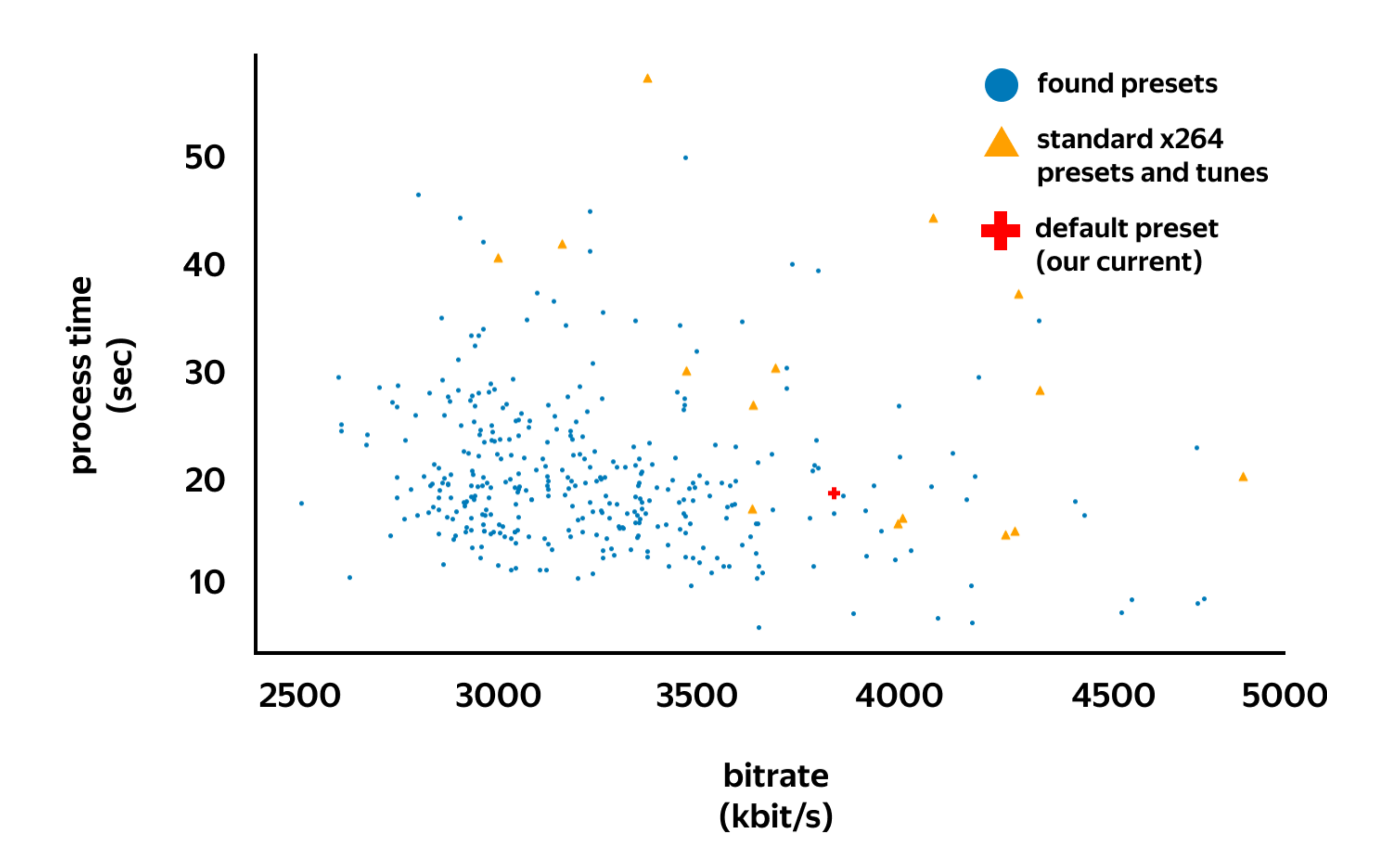
- Итоговый **битрейт**
- Итоговое **визуальное качество** — примерно одинаковые метрики
- **Скорость** кодирования

Дзен



Идея 4: пресеты

Фиксируя **целевое значение** визуального качества, **научились строить** аналогичные картинки



Дзен

57

HighLoad++ 2022

Идея 4: пресеты

Как фиксировать метрику визуального качества?



Идея 4: пресеты

Дзен

Как фиксировать метрику визуального качества?

- Бинпоиском перебираем параметр crf для достижения эффекта



Идея 4: пресеты

Дзен

Как фиксировать метрику визуального качества?

- Бинпоиском перебираем параметр crf для достижения эффекта

Как собирать датасет?

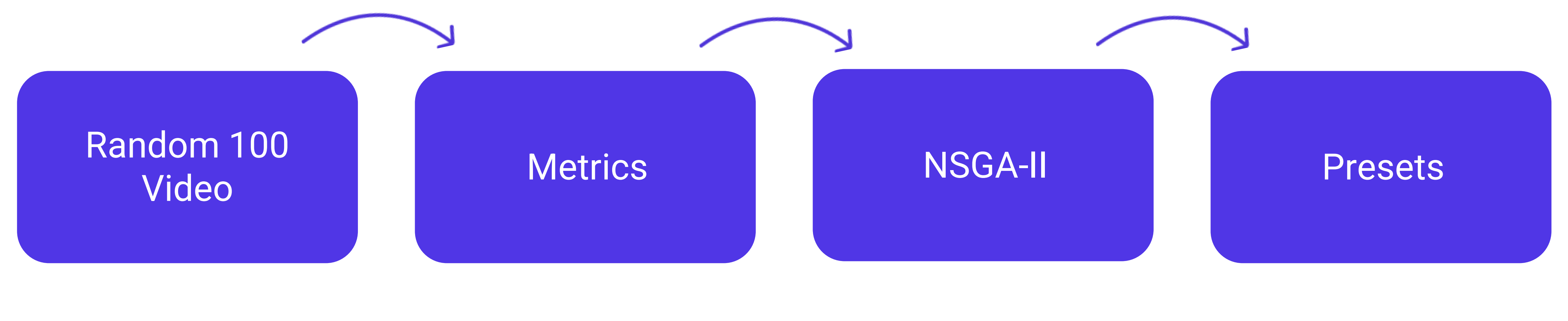


Идея 4: пресеты

Как фиксировать метрику визуального качества?

- Бинпоиском перебираем параметр crf для достижения эффекта

Как собирать датасет?





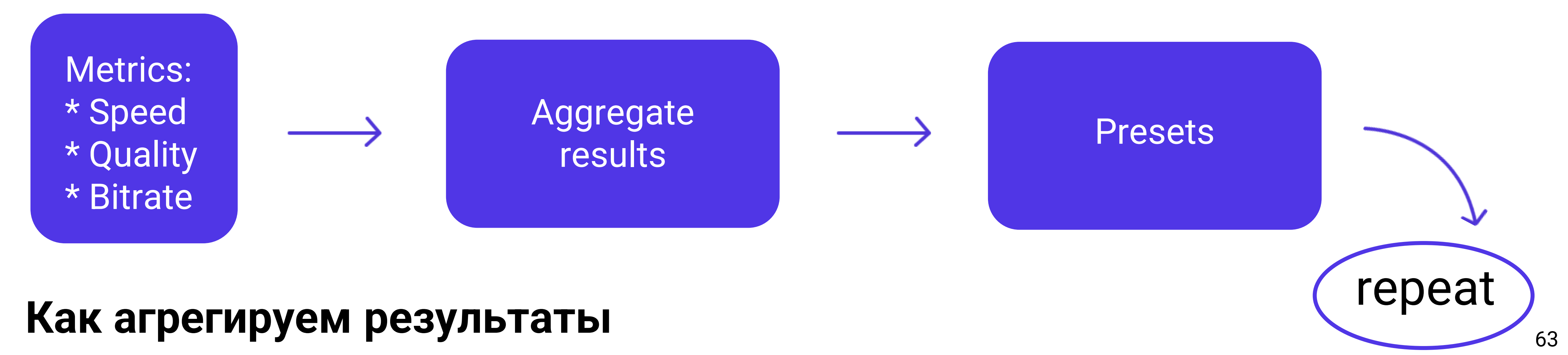
Идея 4: пресеты

Metrics:
* Speed
* Quality
* Bitrate

→ Aggregate results → Presets → repeat



Идея 4: пресеты



Как агрегируем результаты

- Убираем не ок по визуальному качеству
- Убираем не ок по производительности: prod+10%
- Берем топ-1 пресет по битрейту



Дзен

Идея 4: пресеты

Дзен

Metrics:
* Speed
* Quality
* Bitrate

→ Aggregate results → Presets → repeat

Как агрегируем результаты

- Убираем не ок по визуальному качеству
- Убираем не ок по производительности: prod+10%
- Берем топ-1 пресет по битрейту

Итог: ~3000 видео в датасете



Идея 4: пресеты

~150 пресетов собрали по датасету. Это много, хотим меньше





Идея 4: пресеты

Дзен

~150 пресетов собрали по датасету. Это много, хотим меньше

- **Кодируем все пары** video+resolution во все пресеты





Идея 4: пресеты

Дзен

~150 пресетов собрали по датасету. Это много, хотим меньше

- **Кодируем все пары** video+resolution во все пресеты
- Для каждой такой пары определяем **топ-1 пресет**



Идея 4: пресеты

Дзен

~150 пресетов собрали по датасету. Это много, хотим меньше

- **Кодируем все пары** video+resolution во все пресеты
- Для каждой такой пары определяем **топ-1 пресет**
- Собираем **топ-15 частотных** пресетов



Идея 4: пресеты

~150 пресетов собрали по датасету. Это много, хотим меньше

- **Кодируем все пары** video+resolution во все пресеты
- Для каждой такой пары определяем **топ-1 пресет**
- Собираем **топ-15 частотных** пресетов

Итог: датасет Видео+Разрешение+Пресет+Crf



Идея 4: пресеты

Как выбирать оптимальный пресет?



Идея 4: пресеты

Как выбирать оптимальный пресет?

- Кодировать во все пресеты
- Предсказывать. Нужны фичи

Дзен



Идея 4: пресеты

Как выбирать оптимальный пресет?

- Кодировать во все пресеты
- Предсказывать. Нужны фичи

Какие фичи взяли?



Идея 4: пресеты

Как выбирать оптимальный пресет?

- Кодировать во все пресеты
- Предсказывать. Нужны фичи

Какие фичи взяли?

- Фичи, как в подходе с YouTube
- Добавляем новые Ti- и Si- фичи из статьи про пресеты, например, количество деталей в кадре



Идея 4: пресеты





Идея 4: пресеты

- **Получили датасет:** видео+разрешение+фичи+пресет+crf



Идея 4: пресеты

- **Получили датасет:** видео+разрешение+фичи+пресет+crf
- **Обучили модель** на предсказание пресета. Пробовали CatBoost, RF, SVM. SVM оказался лучше с качеством 0.45 f1-score



Идея 4: пресеты

Дзен

- **Получили датасет:** видео+разрешение+фичи+пресет+crf
- **Обучили модель** на предсказание пресета. Пробовали CatBoost, RF, SVM. SVM оказался лучше с качеством 0.45 f1-score
- **Что CRF?** RandomForest модель, для который пресет — фича



Идея 4: что получилось

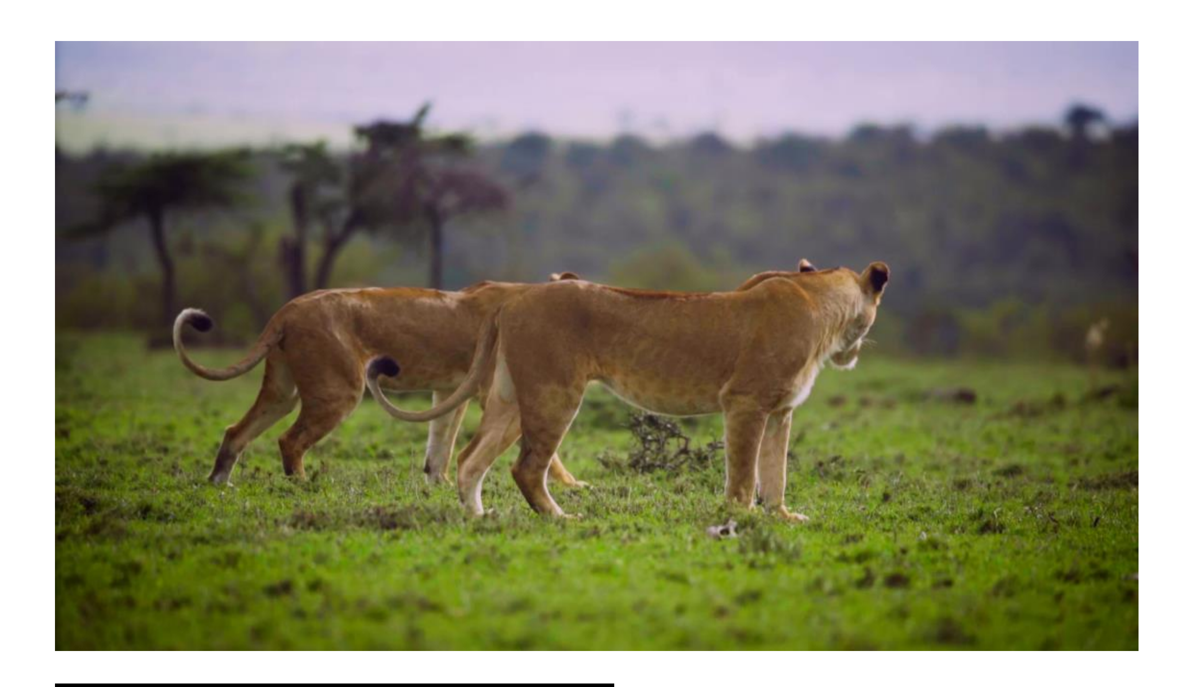
Пример пресета:

```
$ ffmpeg -i src.mp4 -an -c:v
libx264 -crf 20 -level 4.0 -
profile:v high -
force_key_frames
expr:eq(mod(n,120),0) -maxrate
5000k -bufsize 10000k -aq-mode
autovariance -aq-strength 0.5
-psy-rd :0.25 -subq 10 -b-
pyramid strict -deblock -3:-3
-i_qfactor 1.0 -me_method dia
-qcomp 0.5 -rc-lookahead 24 -
refs 1 -trellis 2 -weightp 0
out.mp4
```



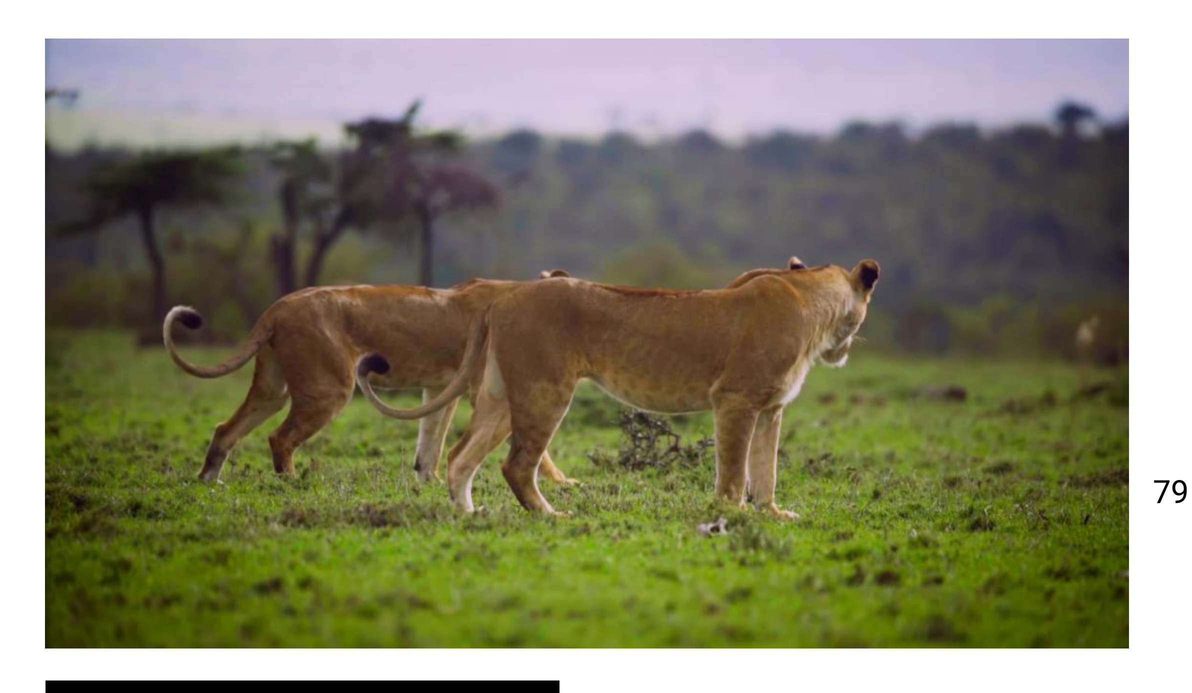
Исходник:
- h264
- FullHD
- 18375 Кбит/с

Идея 4: что получилось



Кодировали раньше:

- h264
- **4787** Кбит/с
- SSIM: **0.9871**
- VMAF: **95.73**

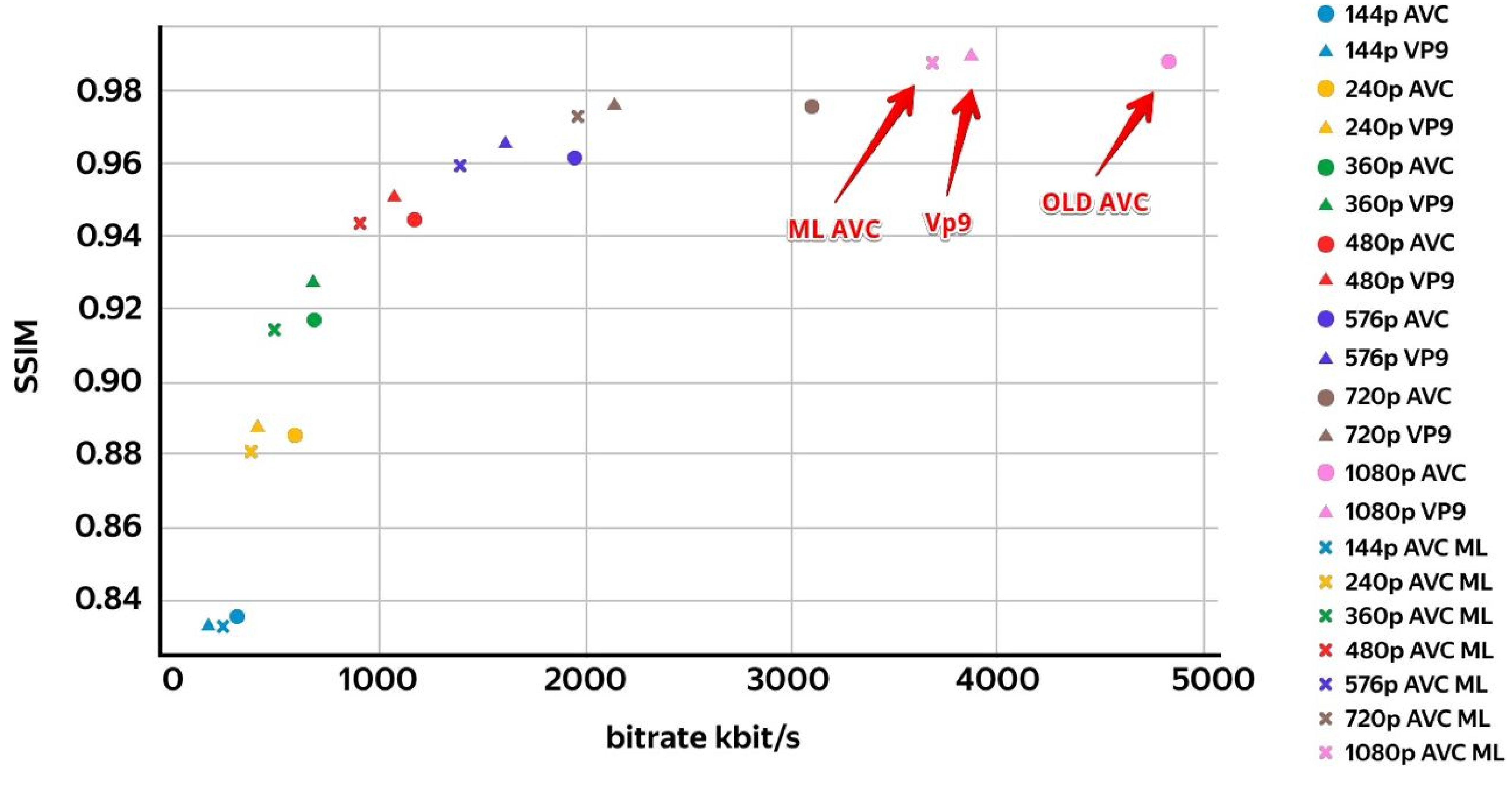


Результат пресета:

- h264
- **3054** Кбит/с
- SSIM: **0.9868**
- VMAF: **96.06**



Идея 4: Результаты



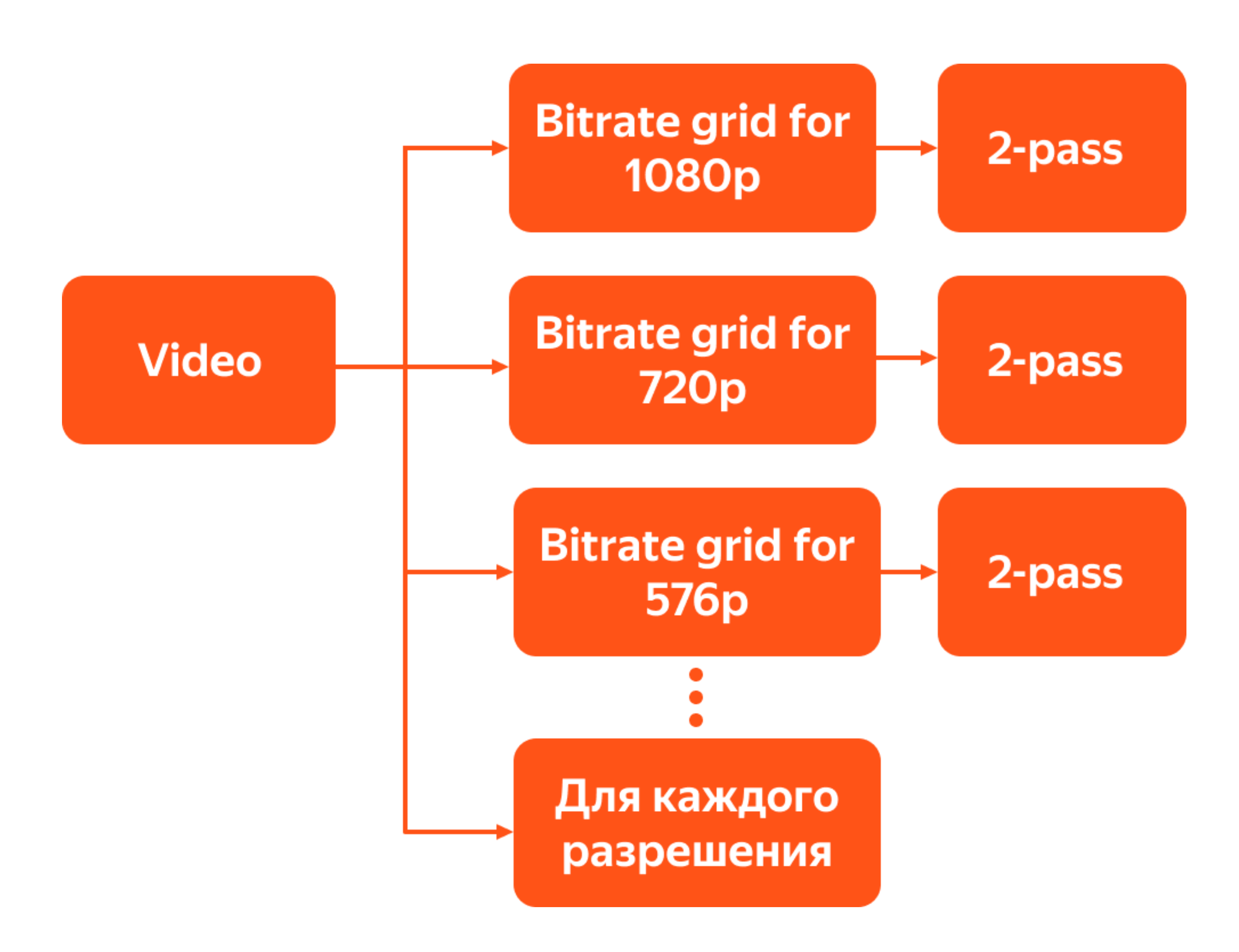
Vp9 проигрывает пресетам h264!

Дзен

HighLoad++ 2022

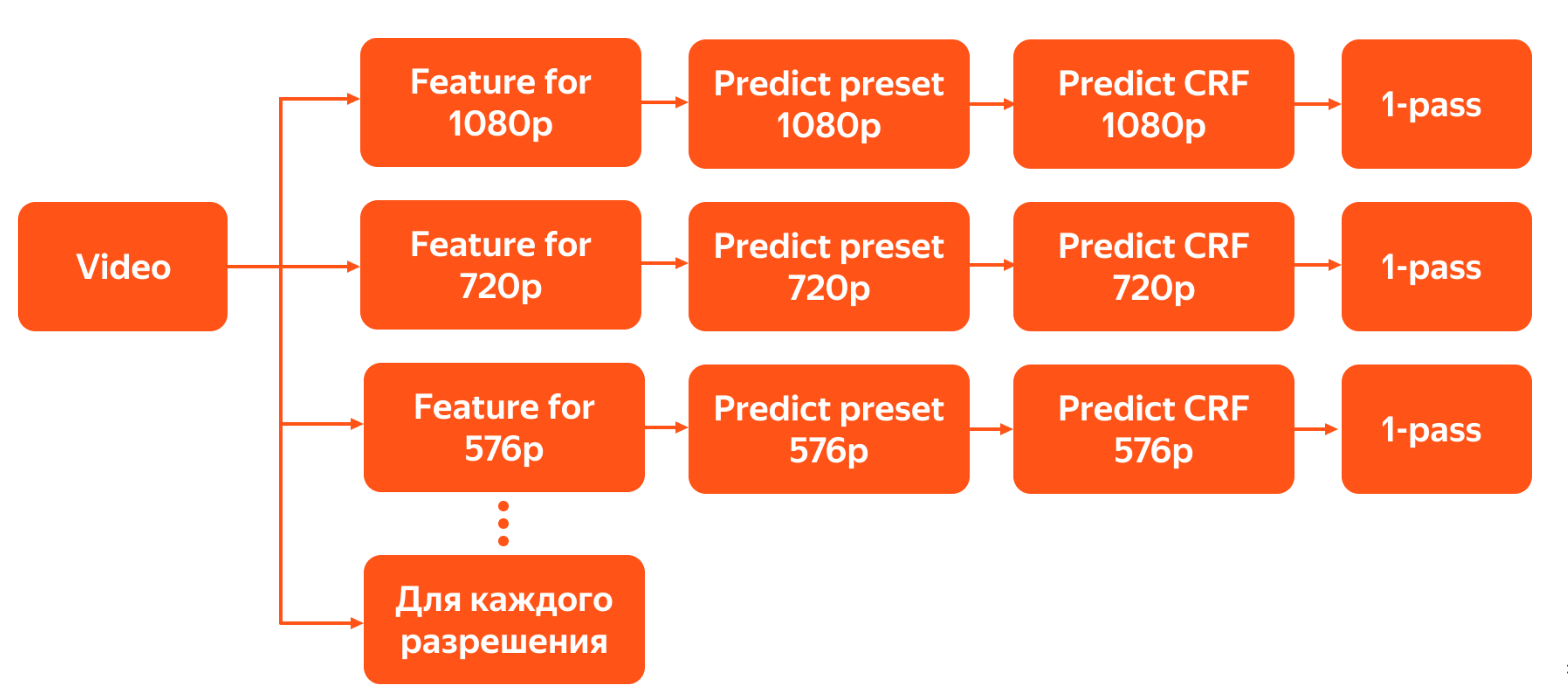
В продакшн!

Как работает транскодер:



В продакшн!

Как стало





576p

720p

1080p

4500

3500

2500

1500

500

Янв 31 Мар 14 Апр 25 Июн 06 Июл 04

Дзен

HighLoad++ 2022

В продакшн!

- **Замедляем** кодирование
- Включили **«в офлайне ночью»** для пережатия топ по трафику
- Как внедрить на всем потоке? **Ускорять!**

Дзен



В продакшн!

- **Замедляем** кодирование
- Включили **«в офлайне ночью»** для пережатия топ по трафику
- Как внедрить на всем потоке? **Ускорять!**

Недостатки:

- Фичи нужно считать для каждого разрешения, а это **еще 5 кодирований**…



В продакшн!

— А можно ли **переиспользовать фичи?**

— Да! **Считаем по 144p** разрешению и переиспользуем в модели



В продакшн!

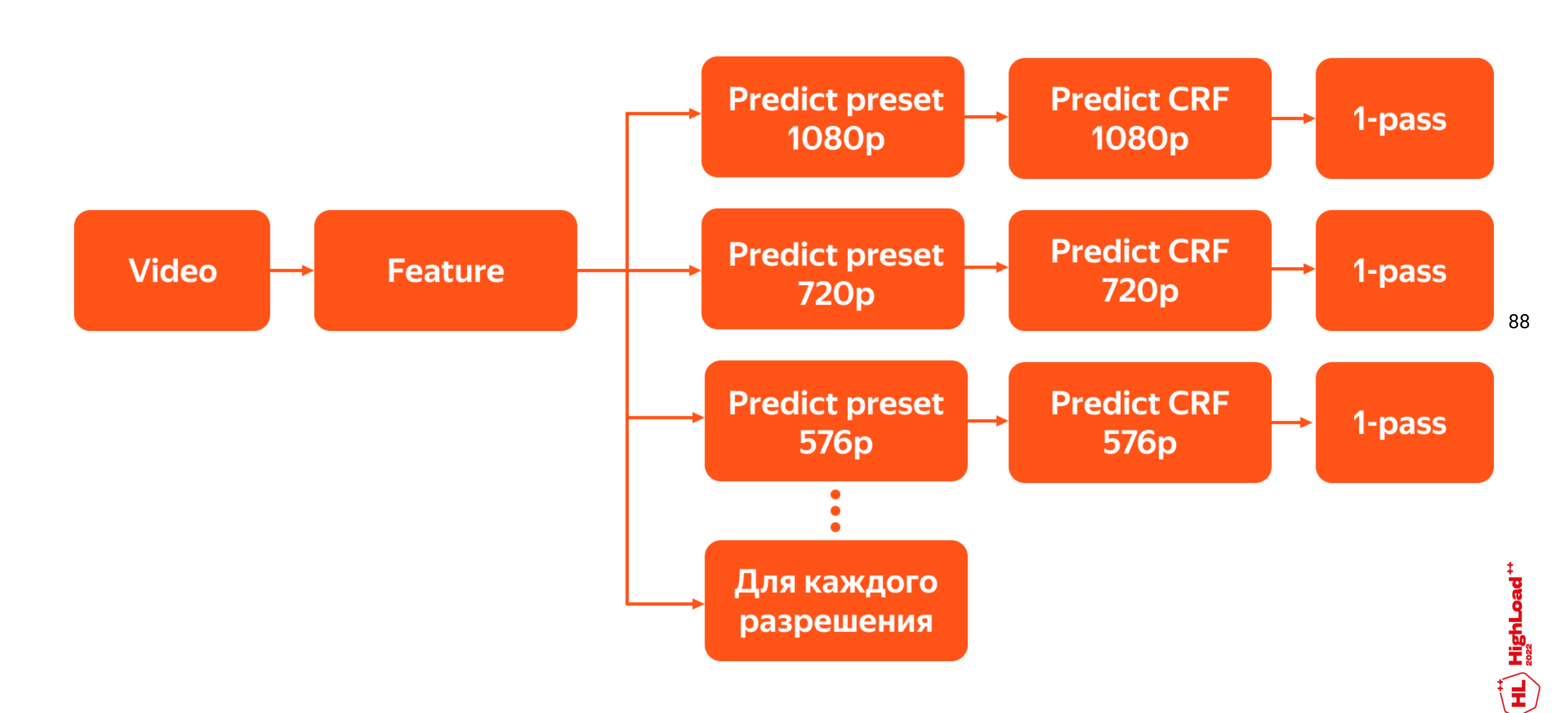
— А можно ли **переиспользовать фичи?**

— Да! **Считаем по 144p** разрешению и переиспользуем в модели

Итог: не просадили скорость кодирования, внедрили на всем потоке



В продакшн!



Video
Feature
Predict preset 1080p
Predict CRF 1080p
1-pass
Predict preset 720p
Predict CRF 720p
1-pass
Predict preset 576p
Predict CRF 576p
1-pass
Для каждого разрешения
Дзен
HighLoad++ 2022
HL++

Результаты

30%
средняя экономия битрейта

до 45%
экономии битрейта на FullHD

Дзен



Что дальше?

Дзен

- H265
- Gpu-кодеки
- Per-scene-кодирование

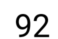


Итог



Итог

- Сделали **экономию в 30%** трафика





Итог

Дзен

- Сделали **экономию в 30%** трафика
- **В тех же условиях:** на том же железе и за то же время



Итог

- Сделали **экономию в 30%** трафика
- **В тех же условиях:** на том же железе и за то же время
- **Оптимизировали** не только трафик, но и **сторадж**



Итог

- Сделали **экономию в 30%** трафика
- **В тех же условиях:** на том же железе и за то же время
- **Оптимизировали** не только трафик, но и **сторадж**
- **Отказались** от кодирования **в vp9**

Дзен



H264 ЖИВ!



Голосуйте за мой доклад
и пишите в TG @baho_emel

